

REMARKS

Claims 13, 17, 19, 23, 26, 31, and 33-35 are pending. Claims 13, 17, 19, 23, 26, 31, 33, and 33-35 are rejected under 35 USC 112 second paragraph. Claims 13, 26, and 33 are rejected under 35 USC 112 second paragraph as omnibus claims. Claims 13, 26, and 33 are rejected under 35 USC 103(a) as being unpatentable over Burgess (US 5,805,896), in view of Sakurai et al. (US 6,334,076), Juras et al. (US 2002/0165744), and Elmqvist ("A Uniform Architecture for Distributed Automation", Advances in Instrumentation and Control, Instrument Society of America, Research Triangle Park, NC US, Vol. 46, Part 2, 1991, Pages 1599-1608). Claims 13, 17, 19, 23, 26, 31, 33, and 34 are rejected under 35 USC 103(a) as being unpatentable over Burgess, in view of Sakurai, Elmqvist, and Leisten et al. (US 6,023,702).

Claims 13, 26, 33, and 35 are amended. Claims 13, 17, 19, 23, 26, 31, and 33-35 are presented for examination. Applicants' paragraph and line numbers mentioned herein are relative to the substitute specification.

Response to 35 USC 112 rejections

Claims 13, 26, and 33 cannot be considered omnibus claims. An omnibus claim reads as follows: "A device substantially as shown and described". This is the definition of an omnibus claim per MPEP2173.05(r), which is nothing like the subject claims. For example, claim 13 has a preamble and 5 clauses, each clause recites one or more limitations, and the claim does not incorporate a specific figure by reference. The drawing is a limitation of the claim and it recites a drawing with the following specific limitations:

- a layout of components of the plant
- based on a material flow
- shows ports with control-relevant information for each component
- least one functional module for each component

Of course examples of such drawings are provided in the application. It is a requirement that patent applications for inventions capable of being illustrated must include drawings. But the claim does not say anything like "a drawing as shown", or "a drawing as shown in FIG 1".

Claims 13, 26, and 33 are amended to recite that the plant layout is input into the system by a user. This is added per Examiner's comment on top of page 4.

Response to claim objection

Examiner misunderstood the terminology of claim 35. The literal values "S" and "P" are not variables, but are the parameter values themselves, as described in paragraph 26 of the specification and as shown in FIG 3. Another way to recite this is "first" and "second" values. Claim 35 is so amended.

Response to 35 USC 103 rejections

The arguments below apply to all of the claims, including the independent claims 13, 26, and 33.

I. Predecessor/successor relationships: The independent claims 13, 26, and 33 recite the generation of plant automation code by graphically mapping input/output information to ports on components based on predecessor/successor relationships contained in a description of the components. Burgess does not include predecessor/successor information in his component objects. Instead his directed relationships are defined in a graphical mapping stage by a programmer.

Examiner asserts on page 5 lines 3-5 of the office action of 05-29-2009 that Burgess in column 2, lines 23-30 discloses sending messages through ports. However, ports are not mentioned in these lines of Burgess, which only abstractly describe message information being passed between event objects.

Examiner asserts on page 5, line 15 that Burgess teaches directed connections of components producing program code, and the developer is shielded from the details thereof. However, this simply means that visual programming can be done, but that is not the issue. The developer still has to define directed relationships graphically, because they are not previously input into the description. Burgess col. 3, lines 29-34, lines 54-57 (FIG 4), and col. 4, lines 1-16 (FIG 5) describe visual programming as illustrated in FIG 4, in which a programmer graphically

configures the directed relationships. Since predecessor/successor relationships are not contained in the descriptions of the objects 410, 420, 450, and 460, nothing prevents a reversal of the C-to-F and F-to-C calculator objects by the visual programmer, which would produce incorrect relationships.

Burgess col. 3, lines 21-38: "FIG. 4 is a diagram illustrating visual programming of the present invention. To generate a visual program to implement a temperature converter, a programmer would position a Fahrenheit scroll bar 401, a Fahrenheit display 430, a Centigrade scroll bar 420, and a Centigrade display 440. The programmer would also position an FtoC calculator 460, which converts a Fahrenheit value to a Centigrade value, and a CtoF calculator 450, which converts a Centigrade value to a Fahrenheit value. In one embodiment, the components are selected from an extendible list of available components. The programmer then connects the components through their ports. The connections 412->461 and 412->431 indicate that when the Fahrenheit scroll bar is changed (e.g., slider moved), the new value is sent to the FtoC calculator and the Fahrenheit display. The connection 462->421 indicates that when the FtoC calculator calculates a new Centigrade value, the new value is sent to the Centigrade scroll bar."

In contrast, Applicants' predecessor/successor relationships are already contained in a description of each component, thus, they are predetermined, constraining the connections to a proper order by allowing fewer degrees of freedom in order to reduce the possibility of error.

Applicants' paragraph 10, lines 1-3: "In the system according to the invention data continuity is achieved in that control-relevant information is already contained in a description."

Applicants' paragraph 20, lines 13-18: "The information already contained in the description 1 is used to allocate input and output information to the ports. The predecessor-successor relationships between the components are governed by this information, i.e. who sends data to whom via which data input is defined thereby."

This prevents incorrect relationships that are possible in Burgess, because the predecessor-successor relationships are defined prior to graphical mapping stage for local plant code generation.

Applicants' paragraph 21: "On the basis of the metainformation, the components 2 are connected to one another by automated means. Particular connections between the components 2 can only be implemented if this is permitted by the constraints described in the metainformation. Automated "wiring" of the components 2, and therefore automatic generation of automation code, are therefore effected."

Applicants' predetermined predecessor/successor relationships restrict freedom in order to reduce complexity in automation programming, because incorrect options are eliminated from consideration. Continuity of expert information and earlier know-how guides and limits the plant automation code developer.

Applicants' paragraph 22, all lines: "The work of the development engineer is greatly facilitated thereby, since fewer degrees of freedom exist as a result of the definition of the metainformation, reducing the possibilities of error. In addition, a continuous information flow is ensured, reducing the loss of already established know-how during the development of the automation system."

Applicants' paragraph 34, lines 2-3: automation code is generated on the basis of existing descriptions 1 of a plant structure.

II. Previously input plant structure and know-how: On page 6, lines 4-7 of the office action of 05-29-2009, Examiner concedes that Burgess does not teach code generated on the basis of a structure of a manufacturing or processing plant and know-how including predecessor/successor relationships previously input into the description. These elements are recited in all of the independent claims of this application, including predecessor/successor relationships, as taught in paragraphs 22-25 of the specification. This is not simply an intended use since it prevents sequence errors.

Sakurai provides standard program modules previously coded by program engineers. These standard modules are independent of plant type (col. 4, lines 3-6). A plant operator inputs a plant operation procedure by keyboard, which includes a time sequential control flow (col. 3, lines 61-63). This results in a selection of standard program source code modules for assembly into a load module for execution in a programmable logic controller (PLC). There is no teaching that the standard program modules include a previously entered description of predecessor/successor relationships for plant components.

Examiner cites Sakurai's sequential flow chart (SFC) and block diagram as evidence of automatic code generation using a drawing of a plant layout with automatic predecessor/successor control as claimed by Applicants. However, the operator of Sakurai must select appropriate program modules using a module identification code, must then input specifications of a plant operation procedure to modify each program module, and must specify

their execution order and interconnections. This is extremely more complex than Applicants' code generation as claimed. Sakurai requires an operator to have a systems level understanding of program modules, load modules, symbolic address resolution, linking, and loading.

Sakurai col. 5, lines 30-41: "The plant operation procedure is inputted to the CRT controller 20 preferably in the form of the sequential flow chart (hereinafter called SFC) 52 and arithmetic block diagram 50, by using the keyboard 5. In the CRT controller 20, while observing CRT 21, an operator inputs an identification code of a module through the keyboard 5 to thereby read the module from the stacker 156 via the system 10C. While observing the read-out plant operation procedure information, the operator enters from the keyboard the program generating information necessary for the generation of a program, i.e., program module execution level and order, signal interconnection or the like."

Sakurai col. 6, lines 49-54: "(3) The interconnection relationship between customized modules are represented not by absolute addresses of a memory storing the modules but by the device numbers defined by various list information. After customized modules are combined and edited as a load module, address translation is carried out."

III. Description in drawing: Examiner concedes on page 6, lines 7-8 that Burgess does not teach components described in a drawing comprising control-relevant information in a manufacturing and/or processing plant. On page 7 of the office action Examiner cites Sakurai col. 2, lines 20-51. However, these lines do not mention a drawing or graphics at all. Examiner also cites Sakurai col. 4, lines 8-22. However, these lines describe visually monitoring of a PLC program while it is executing, in order to verify proper program operation -- not graphically configuring a specification for automatic code generation.

IV. Description based on material flow: Examiner concedes on page 6, line 8-10 that Burgess does not teach control information described in a drawing based on a material flow in a manufacturing and/or processing plant. Examiner then asserts on page 8, lines 1-5 that Elmqvist supplies the claimed feature of drawings with control-relevant information based on material flow in a processing plant, because it is inherent that the physical objects of the plant form the path for the material or fluid flow as shown in the example of the tank system (figures 1-5). However, as in Burgess, this tank system layout only exists after the visual designer has selected the graphic components and placed them in this order. These graphical components do not have

predecessor/successor descriptions stored in them to require an order based on a material flow. Instead, the design module definitions of Elmqvist are purely hierarchical (FIG 2). The first line under VISUALIZATION on page 1605 states: "The complete picture, as seen in a window, is a hierarchical picture according to the module hierarchy." The two tanks of FIG 1 are just instantiations of the same "tank" object. Thus their order in FIG 1 could be reversed -- the same kind of mistake as discussed for Burger above. It so happens that this reversal would not make a difference in FIG 1 of Elmqvist, but this is only because FIG 1 is too simple an example to illustrate an ordered relationship. The examples of Elmqvist and Burgess are both highly simplified, but at least the example of Burgess can be used to show how order is important, which is the case in a manufacturing or processing plant.

V. Predecessor/successor relationships in view of Juras: Examiner concedes on page 6, lines 4-7 that Burgess does not teach system components defined to have predecessor/successor relationships. Examiner then cites Juras, who teaches a process for strategic business planning, including developing quotes, creating business work plans, managing operations; and estimating a total cost of product development. Output of this process is a list of things to do (par. 51, FIG 14). This is a business planning tool, not a code generator for a manufacturing plant. It is unrelated to the present invention, which is not a business plan, but an automation code generator for a manufacturing plant.

Juras [0035]: As shown in FIG. 1, the present invention is a product development process which takes development of a product from an initial Request For Estimate (RFE), through the creation of a product development or business plan to an implementation of the plan for a product launch. The product development process sets up a time schedule for delivering the contracted deliverables to the customer.

Juras [0036]: The product development process uses tools embodied in software which provide a structured process for developing quotes, creating business and functional work plans, and managing operations to design and build manufacturing systems to manufacture and deliver the contracted product to the customer. The tools include a menu structure, plant layout, product development and process flow chart.

This process does not generate software, but generates requests for quotes, bills of material, cost estimates, project planning documents, manpower requirements, lists of deliverable work products, approval steps, and the like. Juras mentions plant layout only once,

in paragraph 36 above, in relation to project planning for plant design. He says nothing about predefined predecessor/successor relationships among elements of a plant or generating software to operate a plant. Juras simply coordinates tasks done by humans, which does not apply to the present invention. Nowhere does he show a plant layout, either by a drawing or other means.

VI. Examiner cites Leisten as teaching predecessor and successor activities in a plant. However, Leisten teaches a method of project management of workgroups. His method has nothing to do with generating automation code for a manufacturing and/or processing plant, automatically or otherwise. His method has nothing to do with defining input/output signal connections between components for generation of automation code. He uses house building as an example of a managed project. His predecessor and successor activities simply refer to a calendar schedule of a human work project. Leisten simply coordinates tasks done by humans, which does not apply to the present invention.

Leisten col. 15, lines 34-49: "The term "Work Process" is used in the following for the full integration of process and project management under the inventive concept. The example is taken from the context of a building enterprise in the building industry, where the building general contractor erects houses according to the requests of an owner."

Leisten col. 15, lines 50-53: "For the process of building the standard house, a project schema 102 is defined which manages the application of resources, as building teams and building equipment within well defined schedule and calendar constraints."

This does not fill the deficiency of Burgess, noted by Examiner on page 17 lines 15-19, that Burgess doesn't specifically teach code generation for a manufacturing and/or processing plant, and that automation code is generated on the bases of a structure of the plant and know how, including predecessor/successor relationship previously input into the description.

VII. M.P.E.P. 2143.03 provides that to establish prima facie obviousness of a claimed invention, all words in a claim must be considered in judging the patentability of that claim against the prior art. If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious.

MPEP 2106 II C:

". . . Finally, when evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 188-89, 209 USPQ 1, 9 (1981) ("In determining the eligibility of respondents' claimed process for patent protection under § 101, their claims must be considered as a whole. It is inappropriate to dissect the claims into old and new elements and then to ignore the presence of the old elements in the analysis. This is particularly true in a process claim because a new combination of steps in a process may be patentable even though all the constituents of the combination were well known and in common use before the combination was made.").

Request for withdrawal of finality

Examiner objected to claim 35 for formalities, but never examined it on the merits. Thus, the examination is incomplete.

Error in claim status

Examiner lists and rejects claim 29. However, claim 29 was canceled in the previous response of 26 February 2009.

(Please proceed to the next page.)

Conclusion

M.P.E.P. 2143.03 provides that to establish prima facie obviousness of a claimed invention, all words in a claim must be considered in judging the patentability of that claim against the prior art. If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious.

As argued above, the proposed combinations of Burger, Sakurai, Juras, Elmqvist, and Leisten do not teach or suggest several features of the present invention recited in the independent claims. These features provide major benefits in safety and continuity of plant automation design and code generation over the prior art. Furthermore, Juras is in a different field, and does not teach any element of the invention. Leisten is also in a different field and does not teach any element of the invention. Therefore, Applicant respectfully requests withdrawal of the 35 USC 103 rejections, and allowance of the present claims.

The commissioner is hereby authorized to charge any appropriate fees due in connection with this paper, including the fees specified in 37 C.F.R. §§ 1.16 (c), 1.17(a)(1) and 1.20(d), or credit any overpayments to Deposit Account No. 192179.

Respectfully submitted,

Dated: July 23, 2009

By: Janet D. Hood
Janet D. Hood
Registration No. 61,142
(407) 736-4234

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, New Jersey 08830